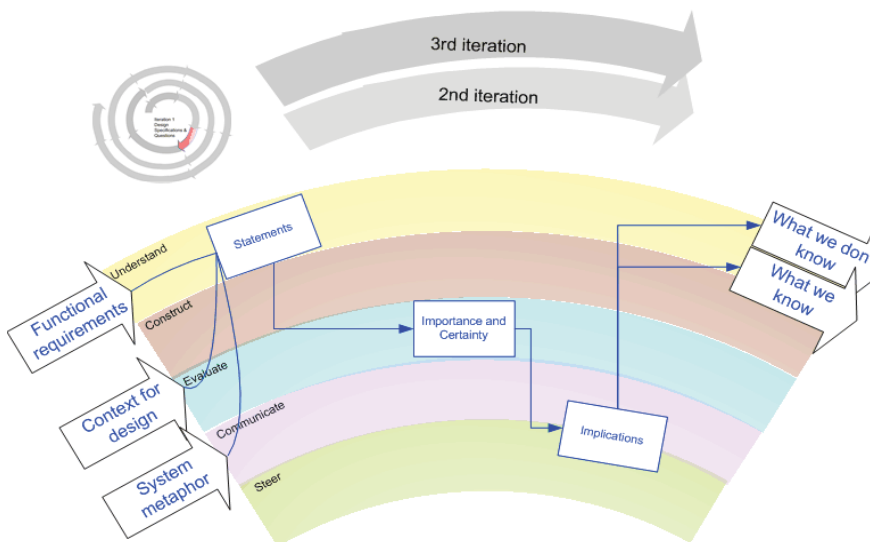


ITERATION 1: DESIGN SPECIFICATIONS



KNOWLEDGE BASE

Bite: Knowledge base statements (with certainty and importance) gives direction for next sector).
Time: 2 hours
Input: Notes from first meeting with client, system metaphor, Ethics
Process evidence: Knowledge base
Client: Discussion

Design specification is about taking the perhaps vague ideas from the design concepts and refining these into detailed blueprints that can be given to programmers.

In the second iteration of the Agile Development Framework the design specifications are the detailed instructions that we can give the programmers (recognising that in many cases this is really us wearing a different hat). There we'll be working towards the first "Functional Delivery" and will be developing detailed screen interface layouts, data structures and system diagrams and so on. Similarly, in the 3rd iteration, where the outcome is the "Robust Delivery", we'll have all of that, plus detailed testing specifications.

So, what do we do in this first iteration? Remember that the purpose of the first iteration is to generate a flow of information in the team (including the client) and to increase understanding about the problem area we're working in. The implementation we're about to do is about building a prototype to help us achieve those things. A prototype, by definition, has a clearly defined purpose: we do it to answer specific questions (rather than a failed first attempt).

This prototype has two main purposes:

1. a quick implementation to use as a model for talking with the client and for users "if we build something like this is it what you had in mind?" , " if we were to build something like this, would it meet your business requirements (/opportunity)?".
2. a chance for the group to get their hands dirty. This is especially important for student projects for two reasons - first it is a chance for the group to see how it operates (is the group member who describes himself as a 'gun programmer' dreaming or a real asset to the group?); secondly it gives the team a chance to experience some of the methods and tools that they might meet when they do the 2nd and 3rd iterations – we would rather find out now that something is going to be particularly tricky to implement than when a delivery deadline is looming.

So, in this design specification sector we are taking everything we know already and recasting it in terms of what we need to know for the first implementation, or perhaps more correctly; in questions that the first implementation can be designed to answer. We do this by creating a "knowledge base". This "knowledge

base” gives us a snapshot of where are and what we know (and what we don’t know) about the project. By going through the project in a structured way, we should already be able to generate a great deal of statements about the project. Some of these we’ll know with a high degree of certainty, some we’ll have a fair idea of, others still we’ll have to make up. This might be a good time to recap a couple of the agile values: courage and honesty.

The questions we ask are all things that we need to know eventually, if not now. The courage value comes in here with recognition that this is your project, no fairy godmother is going to appear with all the answers. Who writes the functional requirements? You do. Who writes the acceptance tests? You do. Who writes the User Manual? You do. (I could play this game all night but you get the picture). The honesty value comes in recognising the certainty and importance we ascribe to each of these statements.

So, you should undertake an exercise to write down everything you know (or don’t know) about the project. Each of these statements should be examined to see how much certainty you can associate with each one. Then, how important they are. A matrix of these measures gives us a steer as to what we should be doing next.

QUESTIONS

This list of questions is intended to prompt you to think about all aspects of the project.

Write answers as stand-alone statements rather than answers to questions.

Don’t be afraid to write other statements that are directly prompted by these questions.

Each question may generate more than one statement

Who is the client?

What is their business? (/what is their mission statement)

What are their key performance indicators?

What is their problem (or business opportunity)?

How did they come to have this business problem/opportunity?

At a high level: how will the project success be measured? (from the client’s point of view/ from yours...)

If the client had unlimited resources, how would you solve the problem?

If the client had already made the decision to employ a person to solve the problem (so your job is to work out that person’s job description) what would that person do ?

What group processes have you put in place (to ensure smooth communication, to resolve differences, to facilitate creativity, to facilitate quality of work...)

Complete the following: “Our evidence portfolio is ... “

Complete the following: “Our Scrum meetings are ... “

What is an appropriate system metaphor for the project?

What is the nature of the benefit to the client? (Involvement in the process; product to sell; enabling new markets; improved business practice etc).

What factors will constrain the development of your project?

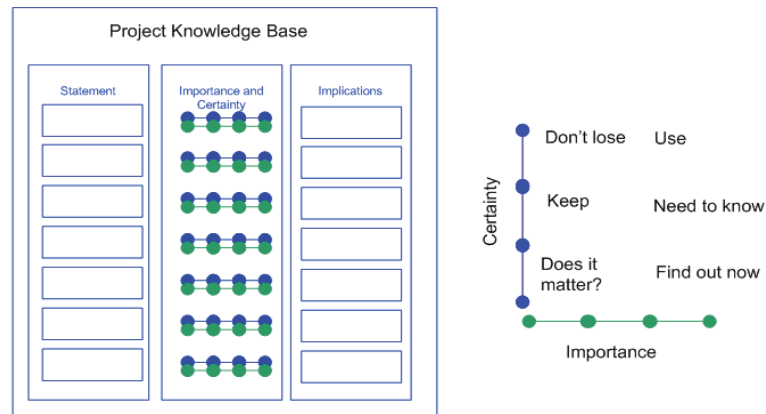
What is the current practice? (what do the people do?, what are the paper systems? what are the computer systems?)

What are the functional requirements? (“The user will ... “)

What are the system requirements for the system?

Who are the users (/what are their characteristics?)

Who are the stakeholders (/what is their relationship with the system, what will they like/dislike about the system)?



If you had unlimited resources, what would you do to solve the problem?

If the instruction was that the problem was not to be solved with computers, what would you do?

The users will really like _____ about this system

The users will really dislike _____ about this system

Has this been done before? Have separate parts been done before?

What are the options on how to acquire it?

What are the costs (not just financial) of this system?

What are the benefits of this system?

Explain the system in ROI (Return on Investment) terms

How would you explain the problem/opportunity to a journalist?

Are there intellectual property issues with this project (infringing on others, or potential for commercialisation)?

Draw a system sketch

What next?

We take the highest ranking statements (least certain and most important) and use them to drive the prototype implementation.

Some questions will not need an elaborate prototype. Others will need answering even before you embark on a prototype. Key questions here are: has this problem been solved before?, what is available off the shelf?, what components could we incorporate?

On presentation of their project to an external industry panel one group got the response

“while we don’t mind if you reinvent the wheel, you should know that you have and make it better; instead, you have reinvented the wheel without reference to the round wheel, and given us a square wheel”.

For example: For a project to develop an “intelligent” robot head for a museum...

TAKING IT FURTHER

The knowledge base could be used as a driver for the whole project. If you were to write the knowledge base on a white board or series of cards (and continually updating it via the Scrum meetings), choosing a task for each day could be as simple as taking the top most ranking statement (most important/least certain) and working to answer it.